

Wprowadzenie

System operacyjny - program, który pośredniczy między użytkownikiem komputera a sprzętem komputerowym

Zadania systemu operacyjnego:

- Ukrywa szczegóły sprzętowe systemu komputerowego poprzez tworzenie *abstrakcji* (maszyn wirtualnych).
Przykłady:
 - jednolity sposób dostępu do urządzeń zewnętrznych
 - zbiory bloków dyskowych widziane jako pliki o symbolicznych nazwach
 - duża, szybka, dedykowana pamięć operacyjna
 - współbieżne wykonanie programów (jako abstrakcja równoległości)
- Zarządza zasobami:
 - zasoby to obiekty niezbędne do wykonania programu, np. pamięć, czas CPU, wejście-wyjście, porty komunikacyjne
 - strategie przydziału i odzyskiwania zasobów (zarządzanie pamięcią, zarządzanie procesorem, zarządzanie plikami, zarządzanie urządzeniami)
 - *efektywność* zarządzania zasobami decyduje o wydajnej eksploatacji sprzętu komputerowego
- Dostarcza przyjazny interfejs
 - *wygoda* użycia (ustawianie przełączników, karty perforowane, taśmy perforowane, terminale graficzne z myszką i klawiaturą)

Składowe systemu komputerowego:

- Sprzęt - podstawowe zasoby obliczeniowe (CPU, pamięć, urządzenia wejścia-wyjścia)
- System operacyjny - nadzoruje i koordynuje posługiwanie się sprzętem
- Programy użytkowe - określają sposób użycia zasobów systemu do rozwiązywania zadań stawianych przez użytkowników
- Użytkownicy (ludzie, maszyny, inne komputery)

Historia rozwoju systemów operacyjnych

Wczesne systemy - „goła” maszyna

- Struktura
 - wielkie maszyny obsługiwane za pośrednictwem konsoli
 - system jednoużytkownikowy (*harmonogram pracy maszyny*)
 - programista/użytkownik pełnił rolę operatora
- Nieefektywne wykorzystanie kosztownych zasobów
 - niskie wykorzystanie CPU
 - pełna sekwencyjność pracy urządzeń
 - przestoje sprzętu związane z wykonywaniem czynności operatorskich
- Wczesne oprogramowanie
asembler, programy ładujące, programy łączące, biblioteki typowych funkcji, kompilatory, programy sterujące urządzeń

Proste systemy wsadowe

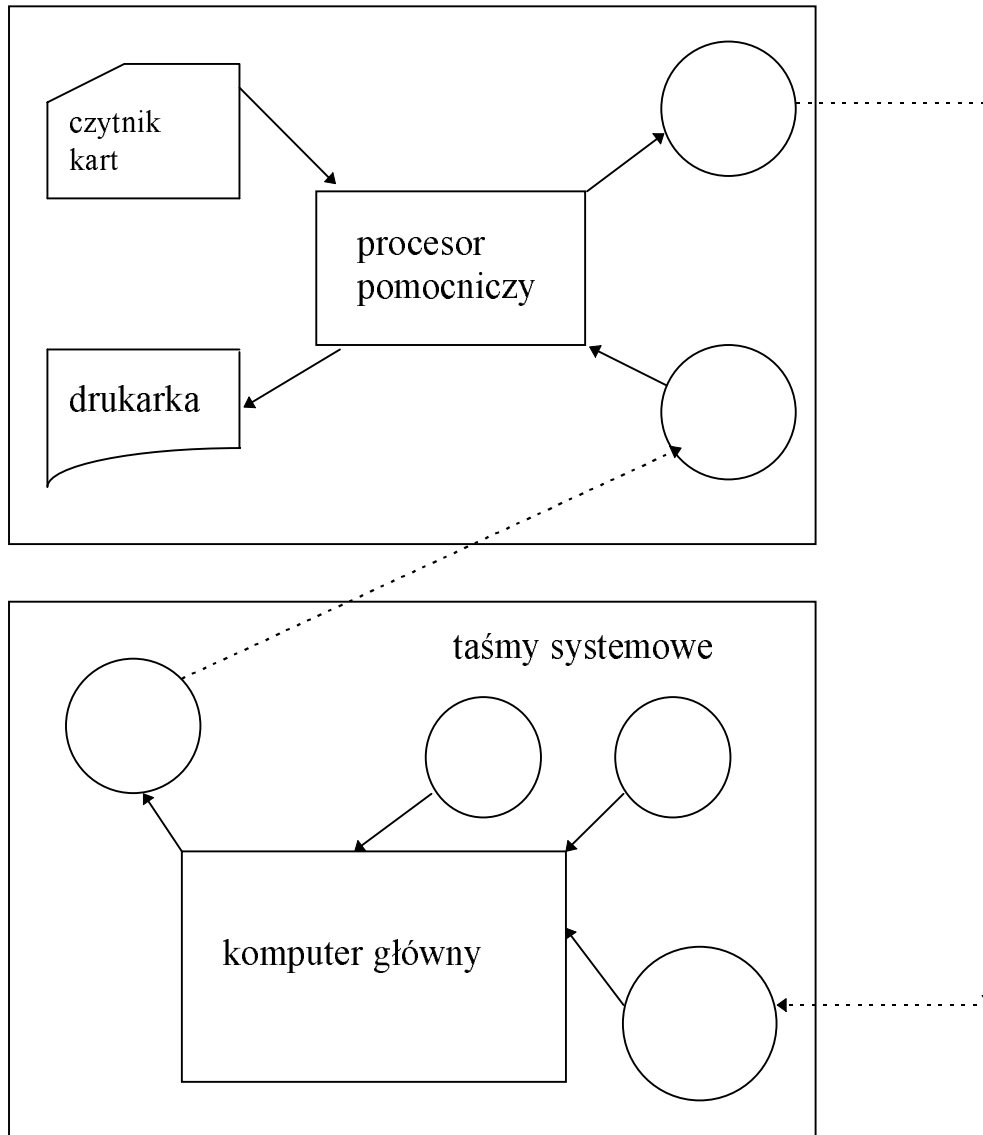
- Zatrudnienie operatora (użytkownik \leftrightarrow operator)
- Skrócenie czasu instalowania zadania przez przygotowywanie wsadu zadań o podobnych wymaganiach
- Automatyczne porządkowanie zadań - automatyczne przekazywanie sterowania od jednego zadania do drugiego
- Rezydentny monitor
 - inicjalnie sterowanie jest w monitorze
 - przekazanie sterowania do zadania
 - po zakończeniu zadania sterowanie wraca do monitora
- Wprowadzenie kart sterujących (*Job Control Language*)
- Istotna zmiana trybu pracy z punktu widzenia użytkownika
- Zwiększona przepustowość systemu kosztem średniego czasu obrotu zadania

Problem: niska wydajność (CPU i urządzenia wejścia-wyjścia nie mogą pracować równocześnie, czytnik kart bardzo wolny)

Rozwiązanie: praca w trybie pośrednim (off-line)

Praca w trybie pośrednim

Przyspieszenie obliczeń poprzez ładowanie zadań do pamięci z taśm oraz czytanie kart i drukowanie wyników wykonywane off-line



Korzyści:

- Komputer główny nie jest ograniczony prędkością pracy czytników kart i drukarek, a jedynie prędkością szybszych stacji taśmowych
- Nie są potrzebne zmiany w programach użytkowych przy przejściu do trybu pracy pośredniej
- Możliwość używania wielu systemów czytnik-taśma i taśma-drukarka dla jednego CPU

Buforowanie

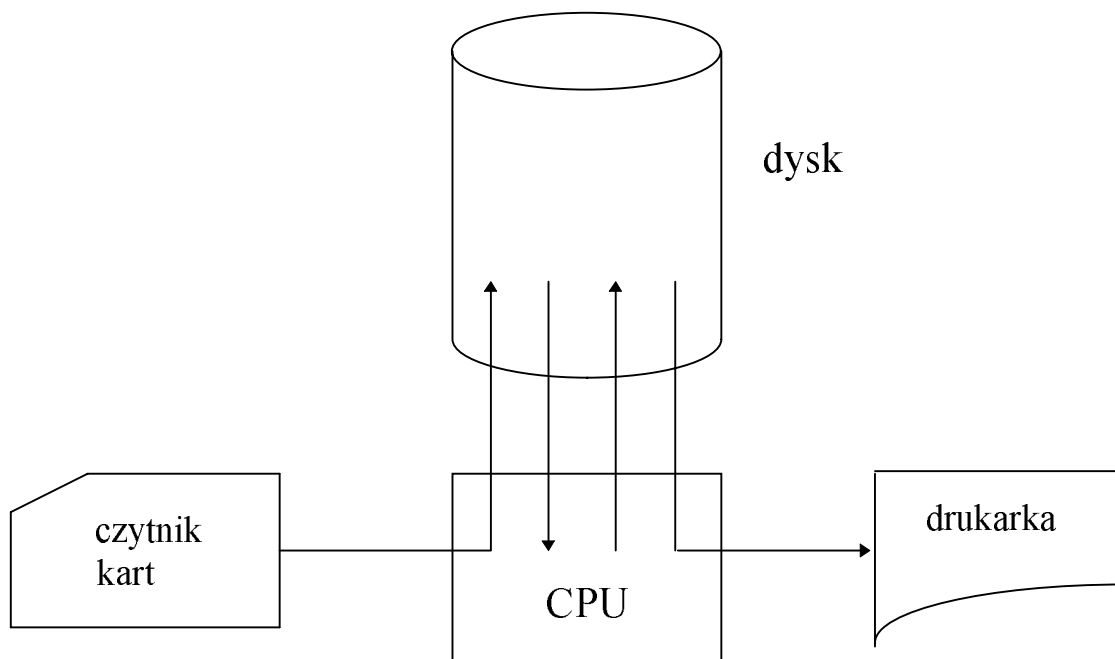
Metoda jednoczesnego wykonywania obliczeń i wejścia-wyjścia dla jednego zadania:

- Nie eliminuje całkowicie przestojów CPU czy urządzeń wejścia-wyjścia
- Wymaga przeznaczenia pamięci na systemowe bufory
- Niweluje wahania w czasie przetworzenia danych

Spooling (simultaneous peripheral operation on-line)

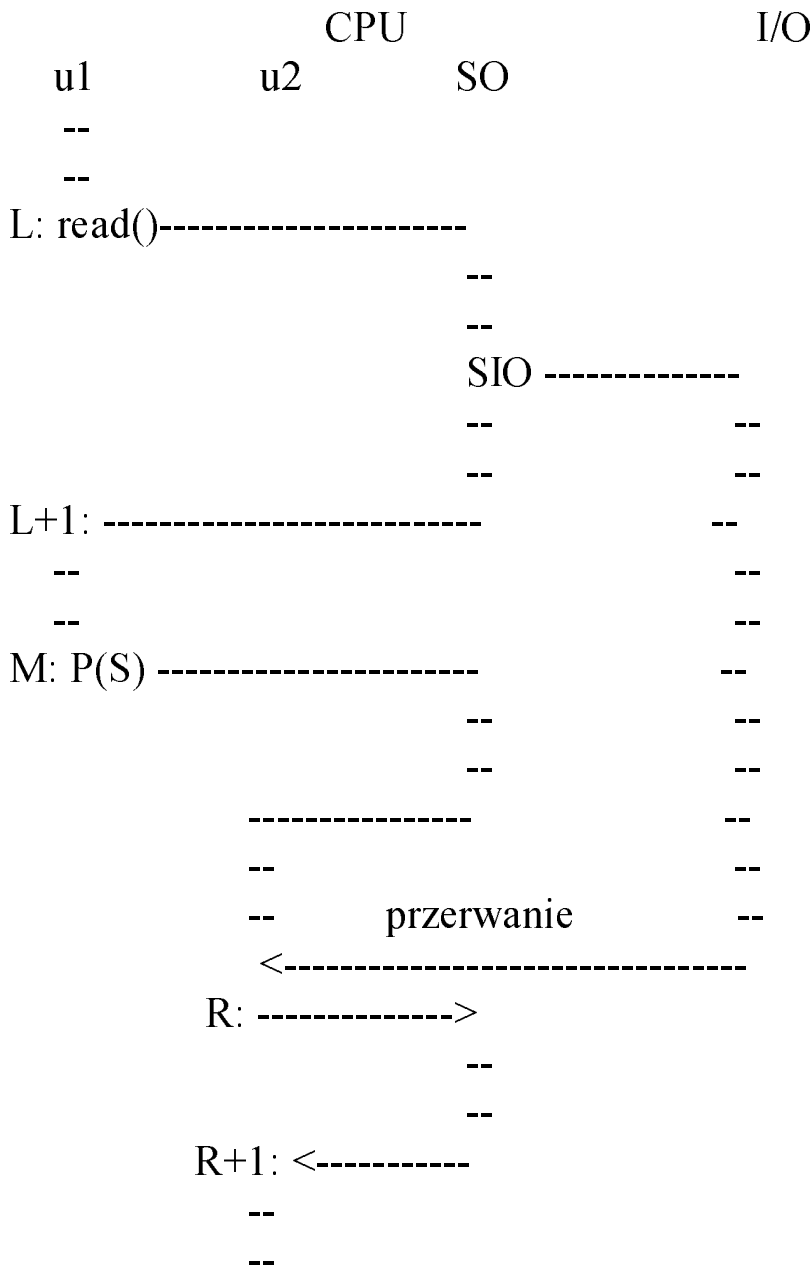
Metoda jednoczesnego wykonywania wejścia-wyjścia jednego zadania i obliczeń dla innych zadań

- Możliwe dzięki upowszechnieniu się systemów dyskowych
- Podczas wykonania jednego zadania system operacyjny:
 - czyta następne zadanie z czytnika kart na dysk (kolejka zadań)
 - drukuje umieszczone na dysku wyniki poprzedniego zadania
- *Pula zadań* - możliwość wyboru kolejnego zadania do wykonania
- Dalsza rozbudowa systemu operacyjnego (moduł wczytujący, moduł sterujący, moduł wypisujący)



Wieloprogramowe systemy wsadowe

Wiele (więcej niż jedno) zadań równocześnie przebywa w pamięci i współdzieli procesor (wykonuje się współbieżnie)



- Dzielenie czasu procesora (wznawianie zadań, synchronizacja i komunikacja)
- Dzielenie pamięci operacyjnej (ile programów równocześnie, ochrona)
- Dzielenie urządzeń (program kanałowy, system przerwań)

Systemy z podziałem czasu

Procesor wykonuje na przemian wiele różnych zadań, przy czym przełączenia następują tak często, że użytkownicy mogą współdziałać z programem podczas jego wykonania

- Interakcja użytkownika z systemem komputerowym (zadanie interakcyjne składa się z wielu krótkich działań)
- System plików dostępnych bezpośrednio (dostęp do programów i danych)
- Wymiana zadania między pamięcią i dyskiem (ang. *swapping*)

Systemy komputerów osobistych

MS-DOS - brak ochrony plików, ochrony pamięci, wielozadaniowości, dualnego trybu pracy

Systemy równoległe

Systemy *wieloprocessorowe* z więcej niż jednym CPU, ścisła komunikacja

- *Systemy ściśle powiązane* - procesory dzielą pamięć i zegar; komunikacja zwykle poprzez pamięć dzieloną
- Korzyści: zwiększona przepustowość, ekonomiczne, zwiększona niezawodność (*łagodna degradacja*)
- Symetryczna wieloprocessowość (każdy procesor wykonuje identyczną kopię systemu operacyjnego, wiele procesów może się wykonywać równocześnie, bez spadku wydajności)
- Asymetryczna wieloprocessowość (każdy procesor wykonuje przydzielone mu zadanie; procesor główny szereguje i przydziela pracę procesorom podrzędnym; częściej występuje w bardzo dużych systemach)

Systemy rozproszone

Rozdzielenie obliczeń pomiędzy wiele fizycznych procesorów

- *Systemy luźno powiązane* - każdy procesor ma lokalną pamięć; procesory komunikują się ze sobą poprzez różne łącza komunikacyjne (szybkie szyny danych, łącza telefoniczne)
- Korzyści: podział zasobów, podział obciążenia, niezawodność, łączność

Systemy czasu rzeczywistego

Służą jako sterowniki urządzeń o ściśle określonym znaczeniu

- Ostre wymagania czasowe
- Sprzężenie zwrotne
- *Przykłady*: sterowanie procesem przemysłowym, monitorowanie stanu zdrowia pacjenta, elektroniczny wtrysk paliwa

Rozwój sprzętu (cztery generacje): lampy, tranzystory, układy scalone małej skali integracji, układy scalone dużej skali integracji

Wymagania sprzętowe:

- system przerwań
- bezpośredni dostęp do pamięci (DMA) dla szybkich urządzeń we-wy
- ochrona pamięci
- wielostanowość procesora
- rozkazy uprzywilejowane
- zegar

Generacje systemów operacyjnych:

- Lata 40. Brak systemów operacyjnych
- Lata 50. Możliwość przetwarzania wsadowego
- Wczesne lata 60. Wieloprogramowość, wieloprocusowość, niezależność urządzeń, podział czasu, przetwarzanie w czasie rzeczyw.
- Połowa lat 60., do połowy lat 70. Systemy ogólnego przeznaczenia umożliwiające jednocześnie różne tryby pracy
- Obecnie. Komputery osobiste, stacje robocze, systemy baz danych, sieci komputerowe, systemy rozproszone

Struktura systemu operacyjnego

Moduły funkcjonalne systemu operacyjnego:

- zarządzanie pamięcią
- zarządzanie procesorem (na poziomie zadań i procesów)
- zarządzanie urządzeniami
- zarządzanie informacją (system plików)
- jądro

Prosta struktura (system monolityczny) - program złożony z wielu procedur, każda może wołać każdą, ograniczona strukturalizacja (np. OS/360, MS-DOS, Unix)

- makrojądro - wszystkie funkcje SO umieszczone w jądrze i wykonywane w trybie uprzywilejowanym (np. OS/360)
- jądro - część funkcji przeniesiona z jądra na poziom użytkownika (np. Unix: jądro i programy systemowe)

Struktura warstwowa - każda warstwa spełnia funkcje zależne tylko od warstwy poniżej (T.H.E., RC-4000, VMS); w Multicsie odmiana struktury warstwowej - zespół koncentrycznych pierścieni. Makrojądro - wszystkie warstwy są uprzywilejowane

Struktura warstwowa systemu T.H.E.:

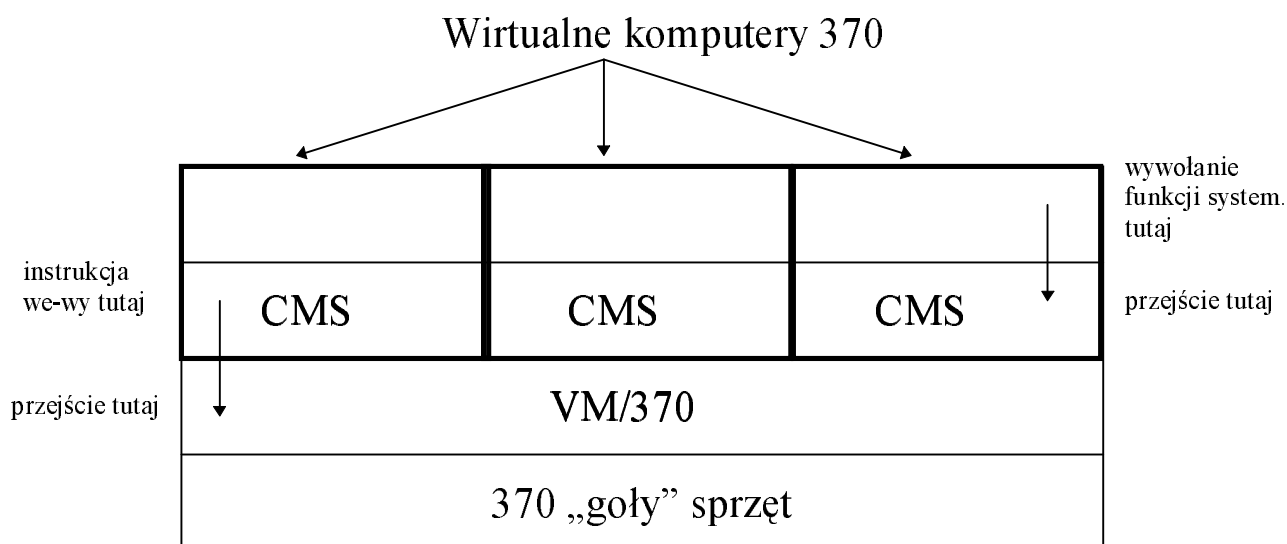
- poziom 5: programy użytkowników
- poziom 4: buforowanie urządzeń wejścia-wyjścia
- poziom 3: program obsługi konsoli operatora
- poziom 2: zarządzanie pamięcią
- poziom 1: planowanie przydziału procesora
- poziom 0: sprzęt

Struktura mikrojądra - mikrojądro obejmuje jedynie tworzenie procesów, komunikację międzyprocesową, mechanizmy (ale nie strategie), zarządzanie pamięcią, procesorem i urządzeniami (na najniższym poziomie), np. Mach

Struktura typu klient-serwer - moduły pełnią wyodrębnione zadania, komunikują się poprzez wysyłanie komunikatów. Serwer - dostarcza usługę, klient - żąda usługi. Prosta adaptacja do systemów rozproszonych.

Koncepcja maszyny wirtualnej: każdy proces otrzymuje (wirtualną) kopię komputera będącego podstawą systemu; każda wirtualna maszyna jest identyczna z rzeczywistym sprzętem, więc każda może wykonywać dowolny system operacyjny; np. VM dla dużych komputerów firmy IBM

Korzyści: dzięki pełnemu oddzieleniu realizacji wieloprogramowości od realizacji „maszyny rozszerzonej” każda część systemu jest dużo mniejsza, bardziej elastyczna i łatwiejsza w utrzymaniu.



CMS - interakcyjny system operacyjny dla indywidualnego użytkownika

Obiektowe systemy operacyjne - system skonstruowany z elementów reprezentujących klientów, odwołania do obiektów, obiekty i zarządców obiektów. Klienci posiadający odwołania do obiektów mogą wykonywać operacje na obiektach implementowanych przez zarządcę obiektów